CLAIMS:

1. A functional random instruction testing (FRIT) method for testing a complex device, comprising:

generating a FRIT kernel;

converting the FRIT kernel into kernel test patterns and storing the kernel test patterns in a tester memory;

loading, at the tester, the kernel test patterns stored in the tester memory onto an on-board memory of a complex device under test (DUT);

executing, at the complex device under test (DUT), a re-generative functional test of the complex device under test (DUT) by applying the kernel test patterns to the complex device under test (DUT); and

comparing, at the tester, a test result of the re-generative functional test with a test expected result to check for manufacturing defects.

2. The method as claimed in claim 1, wherein said FRIT kernel includes a software built-in self-test engine (SBE) configured to execute the re-generative functional test of the complex device under test (DUT), and the expected test result obtained from computer modeling of the complex device under test (DUT).

3. The method as claimed in claim 2, wherein said expected test result is obtained from computer modeling of the complex device under test (DUT) or from a known good device.

4. The method as claimed in claim 2, wherein said software built-in self-test engine (SBE) of the FRIT kernel comprises:

a RIT generator configured with compact RIT machine code that can reside in the on-board memory of the complex device under test (DUT) for generating the re-generated functional test;

a test program execution module configured with test execution directives for providing an environment to store and run the re-generated functional test; and

a test result compaction module configured with compression machine code that compresses test results of the re-generated functional test for storage in the on-board memory of the complex device under test (DUT).

5. The method as claimed in claim 4, wherein said test execution environment employs an exception handler for handling illegal conditions such as undesirable memory accesses, deadlock, shut-down, and infinite loops.

6. The method as claimed in claim 1, wherein said complex device under test (DUT) indicates a microprocessor.

7. The method as claimed in claim 6, wherein, when the kernel test patterns are applied to the microprocessor from the on-board memory, the microprocessor performs the following:

beginning a set-up for executing the kernel test patterns;

executing the kernel test patterns to generate a series of test sequences and associated data for respective test sequences;

running the test sequences, and at the end of the test sequences, obtaining the test results for storage in the on-board memory; and

dumping the test results of the kernel test patterns to the tester, via an interface, for making a comparison with the expected test result to check for manufacturing defects.

8. The method as claimed in claim 7, wherein said FRIT kernel includes a software built-in self-test engine (SBE) configured to execute the re-generative functional test of the complex device under test (DUT), and the expected test result obtained either from computer modeling of the complex device under test (DUT) or from a known good device.

9. The method as claimed in claim 8, wherein said software built-in self-test engine (SBE) of the FRIT kernel is programmed to generate and execute one or more ("N") instruction sequences, each sequence being executed on one or more (M) data sets, where N and M represent

an integer no less than "1" and are user-specified numbers used in generating the FRIT kernel by an especially designed software tool.

10. The method as claimed in claim 9, wherein said software built-in self-test engine (SBE) of the FRIT kernel is further programmed to generate one or more signatures to provide a unique identification of the test result of each test sequence and indicate whether the test result of a particular test sequence is "good" or "bad".

11. A computer readable medium having stored thereon a functional random instruction test (FRIT) kernel which, when executed by a tester system, cause the tester system to:

receiving the FRIT kernel in kernel test patterns and storing the kernel test patterns in a tester memory;

loading the kernel test patterns stored in the tester memory onto an on-board memory of a complex device under test (DUT);

enabling execution, at the complex device under test (DUT), a re-generative functional test of the complex device under test (DUT) by applying the kernel test patterns to the complex device under test (DUT); and

making a comparison between a test result of the re-generative functional test and a test expected result to check for manufacturing defects.

12. The computer readable medium as claimed in claim 11, wherein said FRIT kernel includes a software built-in self-test engine (SBE) configured to execute the re-generative functional test of the complex device under test (DUT), and the expected test result.

13. The computer readable medium as claimed in claim 12, wherein said expected test result is obtained from computer modeling of the complex device under test (DUT) or from a known good device.

14. The computer readable medium as claimed in claim 12, wherein said software built-in self-test engine (SBE) of the FRIT kernel comprises:

a RIT generator configured with compact RIT machine code that can reside in the on-board memory of the complex device under test (DUT) for generating the re-generated functional test;

a test program execution module configured with test execution directives for providing an environment to store and run the re-generated functional test; and

a test result compaction module configured with compression machine code that compresses test results of the re-generated functional test for storage in the on-board memory of the complex device under test (DUT).

15. The computer readable medium as claimed in claim 14, wherein said test execution environment employs an exception handler for handling illegal conditions such as undesirable memory accesses, deadlock, shut-down, and infinite loops.

16. The computer readable medium as claimed in claim 11, wherein said complex device under test (DUT) indicates a microprocessor.

17. The computer readable medium as claimed in claim 16, wherein, when the kernel test patterns are applied to the microprocessor from the on-board memory, the microprocessor performs the following:

beginning a set-up for executing the kernel test patterns;

executing the kernel test patterns to generate a series of test sequences and associated data for respective test sequences;

running the test sequences, and at the end of the test sequences, obtaining the test results for storage in the on-board memory; and

dumping the test results of the kernel test patterns to the testing system, via an interface, for making said comparison with the expected test result to check for manufacturing defects.

18. The computer readable medium as claimed in claim 17, wherein said FRIT kernel includes a software built-in self-test engine (SBE) configured to execute the re-generative

functional test of the complex device under test (DUT), and the expected test result obtained from computer modeling of the complex device under test (DUT) or from a known good device.

19. The computer readable medium as claimed in claim 18, wherein said software built-in self-test engine (SBE) of the FRIT kernel is programmed to generate and execute one or more ("N") instruction sequences, each sequence being executed on one or more (M) data sets where N and M represent an integer no less than "1" and are user-specified numbers used in generating the FRIT kernel by an especially designed software tool.

20. The computer readable medium as claimed in claim 19, wherein said software built-in self-test engine (SBE) of the FRIT kernel is further programmed to generate one or more signatures to provide a unique identification of the test result of each test sequence and indicate whether the test result of a particular test sequence is "good" or "bad".

21. A test system for testing a complex device, comprising:

a complex device under test (DUT) having an on-board memory; and

a tester including a tester memory arranged to test a functionality of the complex device under test (DUT) by:

receiving the FRIT kernel in kernel test patterns and storing the kernel test patterns in the tester memory;

loading the kernel test patterns stored in the tester memory onto the on-board memory of the complex device under test (DUT);

enabling execution, at the complex device under test (DUT), a re-generative functional test of the complex device under test (DUT) by applying the kernel test patterns to the complex device under test (DUT); and

making a comparison between a test result of the re-generative functional test and a test expected result to check for manufacturing defects.

22. The test system as claimed in claim 21, wherein said FRIT kernel includes a software built-in self-test engine (SBE) configured to execute the re-generative functional test of the complex device under test (DUT), and the expected test result.

23. The test system as claimed in claim 22, wherein said expected test result is obtained from computer modeling of the complex device under test (DUT) or from a known good device.

24. The test system as claimed in claim 22, wherein said software built-in self-test engine (SBE) of the FRIT kernel comprises:

a RIT generator configured with compact RIT machine code that can reside in the on-board memory of the complex device under test (DUT) for generating the re-generated functional

test;

a test program execution module configured with test execution directives for providing an environment to store and run the re-generated functional test; and

a test result compaction module configured with compression machine code that compresses test results of the re-generated functional test for storage in the on-board memory of the complex device under test (DUT).

25. The test system as claimed in claim 24, wherein said test execution environment employs an exception handler for handling illegal conditions such as undesirable memory accesses, deadlock, shut-down, and infinite loops.

26. The test system as claimed in claim 21, wherein said complex device under test (DUT) indicates a microprocessor.

27. The test system as claimed in claim 26, wherein, when the kernel test patterns are applied to the microprocessor from the on-board memory, the microprocessor performs the following:

beginning a set-up for executing the kernel test patterns;

executing the kernel test patterns to generate a series of test sequences and associated data for respective test sequences;

running the test sequences, and at the end of the test sequences, obtaining the test results for storage in the on-board memory; and

dumping the test results of the kernel test patterns to the tester, via an interface, for making a comparison with the expected test result to check for manufacturing defects.

28. The test system as claimed in claim 27, wherein said FRIT kernel includes a software built-in self-test engine (SBE) configured to execute the re-generative functional test of the complex device under test (DUT), and the expected test result obtained from computer modeling of the complex device under test (DUT) or from a known good device.

29. The test system as claimed in claim 25, wherein said software built-in self-test engine (SBE) of the FRIT kernel is programmed to generate and execute one or more ("N") instruction sequences, each sequence being executed on one or more (M) data sets where N and M represent an integer no less than "1" and are user-specified numbers used in generating the FRIT kernel by an especially designed software tool.

30. The test system as claimed in claim 29, wherein said software built-in self-test engine (SBE) of the FRIT kernel is further programmed to generate one or more signatures to provide a unique identification of the test result of each test sequence and indicate whether the test result of a particular test sequence is “good” or “bad”.